IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

TITLE OF THE INVENTION

UNIQUE DIGITAL SIGNATURE

INVENTORS

BENJAMIN K. GIBBS

G.R. KONRAD ROEDER

Prepared by

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1026
(303) 740-1980

EXPRESS MAIL CERTIFICATE OF MAILING

I hereby certify that I am causing the above-referenced correspondence to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated below and that this paper or fee has been addressed to the Commissioner for Patents, P.O. Box 1450, Mail Stop Patent Application, Alexandria, VA 22313.

Express Mail No. EV 410001505 US
Date of Deposit: January 27, 2004
Name of Person Mailing Correspondence: April Worley

Signature

1/27/04
Date

UNIQUE DIGITAL SIGNATURE

Cross Reference to Related Applications

[0001] This application is a continuation of U.S. Application Serial No. 09/538,109, filed March 29, 2000, pending, which is a continuation of U.S. Patent No. 6,085,321 and is incorporated herewith.

FIELD OF THE INVENTION

[0002] The present invention pertains to the field of electronic user identification, including, more specifically a unique digital signature.

BACKGROUND OF THE INVENTION

[0003] In the 1960's, the Advanced Research Projects Agency (APRA) of the United States Department of Defense developed and deployed a network of interconnected computers primarily designed to allow research organizations and universities to more easily exchange information. Called the ARPANET, this network of computers was used primarily by the scientific and academic community for research oriented tasks and information exchange. In the 1980's, the ARPANET was replaced by the NSFNET, which is commonly referred to today as the Internet.

[0004] Whereas the Internet was certainly useful in scientific and academic circles, it suffered a serious problem, namely, its interface was difficult to use. In 1989, an English computer scientist named Timothy Berners-Lee introduced the World Wide Web ("WWW"). The WWW was originally designed to facilitate communications over

the Internet between physicists working for the European Laboratory for Particle Physics, but the WWW's ease of use caught on quickly with the both the scientific and non-academic communities. This surge in popularity spurned the development of numerous WWW browsers that enable users to "surf" the WWW.

[0005] Recently, publishers and other information providers have been moving to develop new forms of distribution, similar to traditional "snail mail" subscription services, but on the WWW. Newspapers such as the Wall Street Journal have developed internet sites that offer an alternative to the regular print subscriptions at a significantly reduced cost. Similarly, other periodicals and publications have developed websites for paid subscription subscribers too.

[0006] Today, most pay-for-use subscription sites on the WWW use userid/password pairs that allow a user to logon to a service and review the content in a publication. Although this system works, there are disadvantages. First, during peak hours, the access points (e.g., gateway servers) to the subscription services are often clogged, either by other users trying to logon to the site or general Internet traffic. Second, the userid/password pair gives varying degrees of access to the content of the websites, but access is typically based upon time and levels of content (e.g., all articles or only some sections/services). Third, paying subscribers often share their userid/password pair with other non-paying users. Fourth, because the userid/password pair has a persistent quality (that is, it tends to remain the same over time) unauthorized use (e.g., hacking, snooping, etc.) is common.

[0007] One alternative to the userid/password pair described above is to give access to a user based on the user's computers IP address. Such an alternative is described

in U.S. Pat. No. 5,684,951, invented by Goldman, et al. However, in some computer networks, for example computers connected to a Microsoft's NT DHCP (dynamic host configuration protocol) server, are not given permanent IP addresses. Rather, their IP address varies from session to session. If an IP address is not assigned via DHCP, or a similar dynamic scheme, then it is typically permanently assigned to a particular computer, since multiple computers are generally not allowed to have the same IP address on the same network.

[0008] Beyond the problems associated with the known userid/password pairs, subscribers of a particular service may not wish to purchase a "bundle" of content for a fixed or flat fee. Subscribers may wish to pay for content on a limited use basis and may further desire to pay only for the information they can actually use or specifically request. In light of the foregoing discussion, there is a need for a different method and apparatus for user authentication to an electronic service.

SUMMARY OF THE INVENTION

[0009] The present invention is directed to a unique digital signature comprising a service id and an adapted digital signature. According to one embodiment, the unique digital signature further comprises a domain name.

[0010] According to one embodiment, a unique digital signature is created by an electronic commerce system. The electronic commerce system comprises a router, an authenticated message server and an authentication log file.

[0011] According to one embodiment, a process for creating a unique digital signature comprises the acts of incrementing an index number and hashing the index number and a system key. Next, a value derived from the hash is concatenated with a service id. In one embodiment, the service id is a local username. In an alternative embodiment, the service id is an automated process. Finally, the concatenated value, the unique digital signature, is returned.

[0012] According to one embodiment, an authentication process comprises the steps of extracting an adapted digital signature and a service id from the unique digital signature. In the next step, the service id is tested to ensure it is valid. If the service id is valid, then the adapted digital signature is authenticated, and if the adapted digital signature is positively authenticated, then a status flag is set in a log file to identify the unique digital signature as "used".

[0013] As a result of the method and apparatus described herein, unsolicited or undesired electronic messages can be controlled. Additionally, access to electronic service such as electronic media can be provided on an element-by-element basis, rather than on a fixed period subscription basis.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] The present invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings which like reference numerals refer to similar elements and in which:

[0015] **FIG. 1** is a block diagram illustrating a functional description of the creation of a unique digital signature;

[0016] **FIG. 2** depicts records in an exemplary authentication log file according to presently preferred embodiment of the invention;

[0017] **FIG. 3** is a block diagram illustrating one embodiment of a unique digital signature messaging system;

[0018] **FIG. 4** is a block diagram depicting a preferred embodiment of a unique digital signature electronic commerce system;

[0019] **FIG. 5** is a flowchart depicting the steps of generating a unique digital signature;

[0020] **FIG. 6** is a flowchart depicting the steps of verifying a unique digital signature; and

[0021] **FIG. 7** is a flowchart depicting the present invention as employed in a unique electronic commerce system.

DETAILED DESCRIPTION

[0022] FIG. 1 depicts a functional overview of part of an authenticated message server 100 according to one aspect of the invention. According to a presently preferred embodiment, an authenticated message server functionally comprises a digital signature engine 120. The digital signature engine 120 is fed a service id 104, a system key 108 and an index number 112. The digital signature engine 120 preferably operates on the service id 104, the system key 108 and the index number 112 with a one-way hash function 116. The output of the digital signature engine is a fixed-width binary value referred to herein as a "digital signature" 124.

[0023] According to a presently preferred embodiment, the MD5 function is the one-way hash function 116. The MD5 function is described in detail in RFC 1321, entitled "The MD5 Message-Digest Algorithm", R. Rivest, 1992, which is incorporated herein by reference. In an alternative embodiment, the SHA-1 function is employed.

[0024] Next, an adaptation algorithm 128 performs a base64 conversion of the digital signature 124 and produces an "adapted digital signature" 144. The adapted digital signature 144 is an ASCII value. The adapted digital signature 144 is concatenated, along with other information, to form a unique digital signature 132. The unique digital signature 132 typically comprises three parts: a service id 136, a digital signature 144 and a domain name 148. The service id 104, generally represents an individual or process which should handle an authenticated unique digital signature 132. The adapted digital signature 144 is a special token that contains data that must be authenticated by the authenticated message server 100. Finally, the domain name 148 contains data used by

external routers to forward a message to the appropriate message server connected to the Internet.

[0025] Fixed-width values may be used in the unique digital signature 132 to ease the separation of the fields, or alternatively, additional characters may be added that will identify the three component parts, such as a period ("."), an underscore ("__"), and/or an at sign ("@"). An additional character is preferably concatenated between the service id 136 and the adapted digital signature 144, as well as between the adapted digital signature 144 and the domain name 148.

[0026] According to one embodiment, service id 136 is the same as service id 104. However, according to another embodiment, service id 136 additionally comprises a system key number 140, which is a pointer to the system key 108 used as the input to the one-way hash function 116. Furthermore, it should be noted that service ids 104 and 136 can represent a local username (for example, an alias for an individual user of a message system), or it can represent a local servicename (for example, an alias for an automated process or stored procedure in a messaging system).

[0027] The system key 108 is preferably a 256 bit value that changes periodically and is randomly generated by the authenticated message server 100. Further details of the system key 108 and the index number 112 are explained below with reference to FIG. 2.

[0028] The index number 112 is a counter, maintained by the authenticated message server 100, that identifies one of a number of unique digital signatures generated for a particular system key 108. The index number 112 is periodically reinitialized by the authenticated message server 100 when a fixed number of unique digital signatures 132

have been generated with a particular system key 108, or when a new system key 108 is created.

[0029] In a preferred embodiment, an authentication log file 200, shown in FIG. 2, is used for recording status information and other information about the creation and authentication of unique digital signatures. Since the unique digital signature 132 is preferably transferable, other methods for validating the unique digital signature (i.e., storing remote user information in a log file) will not necessarily work for the preferred embodiment, for example, the methods disclosed and described in U.S. patent application Ser. No. 09/133,875 filed on the same day herewith. However, if a non-transferable unique digital signature is desired, then the methods and techniques described in Ser. No. 09/133,875, which is incorporated herein by reference in its entirety, can be employed.

[0030] FIG. 2 depicts five exemplary records (204, 208, 212, 216 and 220) of an authentication log file 200 according to a presently preferred embodiment of the authentication log file 200. The records of the authentication log file 200 preferably have three fields. The system key number field 230 stores the system key number. Preferably, the system key number 140 (FIG. 1), points to a corresponding record of the authentication log file 200 via the system key number field 230. The system key 108 used by the digital signature engine 120 is stored in system key field 234, and the status field 240 is a bit vector for storing status information about each of the unique digital signatures successfully authenticated by the authenticated message server 100. For simplicity, only 8 bits (241 through 248) of the status field 240 are shown, but hundreds of bits can be used.

[0031] Preferably, the authentication log file 200 comprises at least as many records as are desired for cycling the system key for unique digital signatures. For example, if a five day system key cycle is desired, then at least the five records depicted in FIG. 2 would be sufficient. However, if a 100 day system key cycle period is desired, then at least 100 records should be used. An advantage of the configuration of authentication log file 200 depicted in FIG. 2 is that it is highly compact.

[0032] According to one embodiment, a value of "1" in a bit of bit vector 240 means that a particular unique digital signature has been used. For example, for record 212, six of the unique digital signatures generated with system key number "2" (system key 222 having a value 2E21 FD09 029C 674B.sub.hex) have been used, which is evident by counting the six "1"s in the bit vector 224.

[0033] The authentication log file 200 is used in two respects. First, it is used to store system keys used for the generation of unique digital signatures. For example, each time the index number exceeds the maximum number of bits allowed in the status field 240 (e.g., a particular bit vector 224), then a new system key is created using a secure random number generator. A new record is subsequently added to the authentication log file 200 and in the new record is stored a new system key number (e.g., "5") and the new, randomly generated system key. Of course, all of the bits of the bit vector comprising the status field 240 are reset to "unused" (e.g., "0"), and the index number 112 is reset too. Additionally, a system key number counter is incremented, so the authenticated message server 100 can quickly track the active system key used for creating unique digital signatures.

[0034] It is worth noting that the size, or length of the status field 240 (e.g., bit vector 224) determines the number of unique digital signatures available from a single system key 222. For each system key 222 there exists a block of unique digital signatures that is equal in number to the number of bits in the bit vector 224. Referring again to FIG. 2, exactly 40 unique digital signatures are available. However, if a 64 bit vector 224 were used with 256 system keys, then 16,384 unique digital signatures are possible. As will be apparent with reference to FIG. 6 (described in detail below), there is a practical limit on the number of bits in bit vector 224.

[0035] According to one embodiment, system keys are periodically recycled. Thus, it is possible for a unique digital signature to expire before it is used. This will allow an authenticated message server administrator the flexibility to control the ultimate size of the authentication log file 200, as well as the duration of the unique digital signature's validity. Preferably, however, there are more records in authentication log file 200 than there are cycles in a system key cycle period.

[0036] Before a detailed description of the process for generating and authenticating a unique digital signature is presented, a description of embodiments of a system using a authenticated message server 100 is helpful.

[0037] FIG. 3 depicts a unique digital signature messaging system 300 according to one embodiment of the present invention. System 300 includes a server 308, coupled to a terminal unit or personal computer 304, a router 312, an authenticated message server 316 and an authentication log file 318.

[0038] The interconnection or coupling mechanism between the various connectors on the devices of the unique digital signature system 300 is preferably a fiber optic network cable, but it can also be a twisted pair, or a wireless interconnection. According to one embodiment, server 308 is a Sun Microsystems SPARC™ system running electronic message software such as Oracle Corporation's InterOffice™ messaging server. Router 312 is a commercially available Internet router such as a Cisco Systems 7500 Series router. Authenticated message server 316 can run on a standard personal computer, such as an Intel Pentium™ based microprocessor system. However, authenticated message server 316 is alternatively part of the software component stack added to server 308. In such an embodiment, an application programming interface ("API") for the messaging server 308 is added which provides access to the authenticated message server services. Authenticated message server services include generating and authenticating unique digital signatures as described herein. The unique digital signature system 300 can be highly distributed, wherein incoming and outgoing messages are handled by separate servers or computer systems on an interconnected network (e.g. a LAN).

[0039] According to one embodiment of the authenticated message server, a web-based administrator interface is maintained for system configuration, maintenance of subscriber profiles, and retrieval of any log files (e.g., failure log files, authentication log files and message log files). The web-based administrator interface assists in maintaining the unique digital signature system in general since it is possible that the authenticated message server is highly distributed.

[0040] Although authentication log file 318 is depicted as a separate element in FIG. 3, it can be embedded into the authenticated message server 316 or the server 308. Further, according to one embodiment, authentication log file 318 is a database, such as Microsoft Corporation's SQL Server™ or Oracle Corporation's Oracle8™.

[0041] From the server 308, outgoing messages (including electronic media) for transmission to remote users are passed through an internet gateway router, such as router 312. Router 312 is preferably connected to the Internet via a T1 pipeline, or other leased line. Conversely, messages from the Internet to a particular local user or service associated with the server 308 will be passed through router 312.

[0042] A remote user typically resides on a personal computer, such as laptop 332, which is connected to a second server 328. Server 328 is configured similar to server 308, but it can also be a different type of server, such as a Digital Equipment Corporation VAX/VMS™ system. The server 328 is likely to run a different messaging system from that run in server 308. For example, the server 328 may run the University of Washington PINE™ messaging system. Similar to router 312, router 324 is connected to server 328 and the Internet 320.

[0043] Whereas FIG. 3 depicted a unique digital signature system as employed in an electronic messaging application, FIG. 4 depicts a preferred embodiment of a unique digital signature system 400 as employed in an internet server/electronic commerce application. System 400 comprises a computer 404, a message server 408, a message router 412, a proxy server 420, a WWW server 424, an authenticated message server 428, and an authentication log file 432. Components 404, 408, 420, 424 and 428 are shown as interconnected through local area network ("LAN") 436. Authentication log file 432 and

message router 412 are depicted as having separate connections 416 to authenticated message server 428 and message server 408 respectively; however, in another embodiment, they too are connected through the LAN 436. Furthermore, authentication log file 432 can be embedded in authenticated message server 428 and authenticated message server 428 can be a part of the software component stack of the message server 408. Additionally, the message server 408 can have integrated therein the functionality of message router 412. Off-the-shelf hardware and software components, similar to those described with reference to FIG. 3, are used for like components depicted and described with reference to FIG. 4.

[0044] Unique digital signature system 400 is connected to the Internet 444 via a leased line (e.g., a T1 line.) Leased line 440 is shown functionally divided into three separate lines connected to WWW server 424, proxy server 420, and message router 412, even though only one line is used.

[0045] Functionally, WWW server 424 hosts a website with an electronic commerce application and, preferably, an interface (e.g., Perl, CGI, HTML, Java, ASP, ODBC, etc.) to authenticated message server 428. According to one embodiment, WWW server 424 is preferably a Sun Microsystems SPARC™ system, running WWW/Internet server software from Netscape Corporation. A remote user, for example a user on laptop 452, which is connected to the Internet 444 via an internet access provider ("IAP") or local area network ("LAN") 448, is typically connect to WWW server 424 through a dedicated communications port over the Internet 444. Once connected, the remote user at laptop 452 can either purchase a unique digital signature 132, or request a particular piece

of media or service from the unique digital signature system 400 using a unique digital signature 132 and the interface on WWW server 424.

[0046] According to one embodiment of the unique digital signature system 400, a request for service is received by the authenticated message server 428 via an electronic message (e.g., e-mail) to a service id (e.g., a user or an automated process on computer 404) that passes through message router 412 and message server 408. Similarly, outbound media (in response to a request for service) is returned to the remote user that requested service via message server 408. Outbound media includes, but is not limited to: ASCII text, HTML files, Java applets, WAV files, AVI files, MPEG files and the like. In one embodiment, message server 408 is a wireless short message/paging service ("SMS"), which includes an e-mail to SMS gateway (referred to hereafter as an "SMS gateway") such as one available from Omnipoint Corporation.

[0047] According to another embodiment, a service provision server (not shown) is employed in the unique digital signature system 400. The service provision server receives requests from the WWW server 424, and passes them to another server, such as authenticated message server 428. The service provision server functionally acts as an intermediary between the WWW server 424 and the authenticated message server 428, passing requests for unique digital signatures, and requests for service that contain unique digital signatures to the authenticated message server 428 for processing. Physically, the functionality of a service provision server can be added to the WWW server 424 and the authenticated message server 428, rather than residing on a single machine.

[0048] The proxy server 420 is not necessary in unique digital signature system 400, but its use is desirable in larger or heavily used electronic service applications.

Proxy server 420 is essentially a high-speed cache for one or more internet servers (e.g., WWW server 424) connected to the LAN 436. Functionally, the proxy server 420 strips off the prefix of any URLs received and compares the remaining URL to datafiles stored in its cache. If there is a match, then, rather than requesting the datafile from the WWW server 424 (which is generally more expensive in terms of processing time and I/O), the cached copy I/O on the proxy server 420 is spooled out to the requester, thereby saving a disk I/O and time. According to one embodiment, the proxy server 420 receives a request for service (e.g., send media to a remote user) after the unique digital signature contained in a request for service has been authenticated. In one embodiment, the proxy server 420 runs on a Sun Microsystems SPARC™ system running proxy server software such as NetCache™ from Network Appliance, Inc. in Santa Clara, Calif. (www.netapp.com).

[0049] As may be apparent from the drawings and description above with reference to FIG. 3 and FIG. 4, the authenticated message server essentially becomes a gatekeeper for providing access to electronic services.

[0051] Turning now to FIG. 5, it depicts the steps for generating a unique digital signature according to one embodiment of the present invention. The following description is made with reference to FIGS. 1 and 4, described above.

[0053] Beginning with step 504, a request for a unique digital signature 132 is received by authenticated message server 428. Typically, this request identifies a particular service id 104 for which a unique digital signature 132 is desired. The request typically comes from either the message server 408 or the WWW server 424, however, it can also come from a client application running on a local (e.g., 404) or remote (e.g., 452) user's personal computer. When the request is received, the index number 112 is

incremented by one unit in step 508. Next, in step 512, the index number 112 is compared against the maximum number of bits available in the status field 240 (e.g., bit vector 224). If the value of the index number 112 exceeds the number of bits in the status field 240, then the process continues to step 536, otherwise the process continues to step 516, described below.

[0054] In step 536, the system key number (not the system key 108) used by the authenticated message server 428 is incremented by one. Next, a new system key 108 is randomly generated at step 540. At step 544, the new system key 108 is stored in the next record of the authentication log file 432 (e.g., record 216, in other words, the record identified by the new system key number). The corresponding status field 240 for the next record is reset so as to indicate that none of the unique digital signatures available for the new system key 108 have been used. In step 548, the index number 112 is reset to "1" and the process continues to step 516.

[0055] In step 516, a hash function, preferably a one-way hash function 116, such as the MD5 function, is performed on the system key 108, the service id 104 and the index number 112 by digital signature engine 120. In an alternative embodiment, the SHA-1 hash function is used at step 116. In step 520, the digital signature 124 generated at step 516 is converted from a binary value to a value acceptable for electronic messaging, i.e., ASCII text, by adaptation algorithm 128. The result is an adapted digital signature 144. Preferably, a base64 conversion is performed on coterminous 6 bits of the hash value by adaptation algorithm 128, then the result is concatenated at step 524 with domain name 148, service id 136 and any other characters, values, or symbols used to identify the fields/parts of the unique digital signature 132 (e.g., "@", "." or "__ ").

However, other symbols are not necessary if fixed-width/character service ids 136 and adapted digital signatures 144 are used.

[0056] Step 520, it should be noted, is not necessary in some applications. The functionality of the adaptation algorithm 128 can be incorporated into the digital signature engine 120 either physically (e.g., with logic components), or by way of the particular hash function 116 used. Furthermore, the adaptation algorithm 128 is not necessary if, for example, the electronic message system supports non-ASCII values. However, according to one embodiment, a limited character ASCII set is used since remote users on legacy electronic message and existing telephone systems can still type the unique digital signature without special software (or hardware).

[0057] According to one embodiment of the unique digital signature 132, system key identification information is added to service id 104, such as the system key number 144, so that the system key 108 used to generate the unique digital signature 132 can be quickly identified. The result is service id 136. In an alternative embodiment, the system key identification information is added to the adapted digital signature 144.

[0058] The unique digital signature 132 is returned to the requestor at step 528 (e.g., via an electronic message from message server 408, or an interface on WWW server 424) and the process ends. In one embodiment, the unique digital signature 132 (or a block of unique digital signatures) is returned the remote user (e.g., a user on laptop 452) as an update to a cookie log in a WWW browser (i.e., Netscape's Navigator 4.0.) Turning now to FIG. 6, it depicts the steps for verifying a unique digital signature 132. In step 604, the unique digital signature is received by the authenticated message server 428. The component parts of the unique digital signature 132 are extracted in step 608 (e.g.,

the service id 104 and the adapted digital signature 144.) In step 612, the service id 104 is tested to verify that it represents a valid local username or servicename. If the service id 104 is not valid, then the request for service is rejected at step 656. If the service id 104 is valid, then processing continues to step 616.

[0059] Note that the unique digital signature 132 can also be rejected based upon the system key number 140. For example, a particular system key can expire due to the lapse of time (e.g., all keys with system key numbers less than "2" can be rejected automatically because of their creation date) or because those system keys have been flagged as "used", or both. According to one embodiment, the above described test is performed before step 620. However, according to another embodiment, the above described test is performed after step 608.

[0060] In step 616, the system key 108 is retrieved from the authentication log file 432. According to one embodiment, the system key number 140 was identified in the service id 136, which was extracted at step 608. Once the system key 108 is located, the index number 112 is set to one in step 560. In step 624, the index number 112 ("I") is tested to determine whether it is less than or equal to the maximum number of bits available in bit vector 240 ("MAX I"). If the index number 112 is less than or equal to MAX I, then processing continues to step 628, otherwise the process continues to step 656.

[0061] Next, in step 628, the index number 112 is tested to determine whether it has already been used. The test at step 628 is performed by testing a bit in bit vector 224, the bit being identified by the index number 112 and the system key number 140. For example, bit 241 of bit vector 224 in record 212--identified by index number 112 having

a value of "1" and system key number "2"--in the authentication log file 432. If the index number 112 is marked as "used", then processing continues to step 652, wherein the index number 112 is incremented by one and the process returns to step 624. Otherwise, processing continues to step 632.

[0062] According to an alternative embodiment, step 624 is not performed before step 628. Rather, step 624 is performed after step 652. Either way, step 624 flows to step 628 or step 656, depending on the outcome of the test at step 624.

[0063] With the system key 108, the service id 104 (which can be only a portion of the service id 136) and the index number 112, the digital signature engine 120 performs a one-way hash function 116 at step 632. In an alternative embodiment, the service id 104 is not an input to the one-way hash function 116. Next, in step 636, the resulting hash value (digital signature 124) is converted from a first digital format, to a second digital format (e.g., binary to ASCII) using an adaptation algorithm 128 (e.g., a base64 conversion). The resulting value, an adapted digital signature 144, is compared against the adapted digital signature in the incoming unique digital signature at step 640. If the two values match, then processing continues to step 644, described below. If the two values do not match, then processing continues to step 652, described above.

[0064] In step 644, the corresponding bit of bit vector 224 is marked as used. For example, bit 241 of bit vector 224 in record 212 is set to "1". Finally, in step 648, the incoming request is accepted and notification of the successful authentication is returned. The authentication process then ends, however, other processing/handling of the request for service continues--depending on the particular application where the unique digital signature is employed.

[0065] FIG. 7 depicts the steps of an electronic commerce or service system wherein the methods and techniques described herein can be employed. In step 704, a unique digital signature 132 is allocated. Preferably, the unique digital signature 132 is allocated based upon the steps depicted in the flowchart described above with reference to FIG. 5. However, additional steps can also be involved, such as the collection of money to purchase the unique digital signature 132. After step 704, a remote user in possession of a unique digital signature 132 may wish to purchase something with the unique digital signature 132. If so, then the remote user sends a request addressed with the unique digital signature 132, which is received by the authenticated message server 428 and authenticated in step 708, preferably by performing the authentication process described above with reference to FIG. 6.

[0066] According to an alternative embodiment, WWW server 424 (FIG. 4) prompts the remote user for access to any unique digital signatures stored in a cookie file on laptop 452. If the remote user authorizes the WWW server 424, the WWW server 424 retrieves a unique digital signature 132 from the cookie file.

[0067] If the unique digital signature 132 cannot be validated, that is it is rejected by the authentication process "A", then an error message is returned to the requestor. If the unique digital signature 132 is validated by the authentication process "A", that is it is accepted, then processing continues to step 712. In step 712, the successfully validated unique digital signature 132 is forwarded to a particular username or automated process (servicename). Next, in step 716, the request identified by the unique digital signature 132 is processed by a stored procedure, or by a local user, as the case may be. Finally, at step 720, a response message is returned to the remote user of the unique digital signature

132. According to one embodiment, media files are returned, such as Java applets, one or more bundled HTML files, an MPEG file, a WAV file, or a RAM file. In another embodiment, the media files are stored on a proxy server 420 and accessed at the proxy server 420 by the remote user.

[0068] For example, one application where a unique digital signature 132 can be employed is in an electronic voting or polling system. In such a system, the unique digital signature 132 is allocated by the authenticated message server 428 at step 704. In one embodiment, the unique digital signature 132 is sent to a remote user (e.g., 452) via WWW server 424. In another embodiment the unique digital signature 132 is sent via electronic message server 408. In still another embodiment, the unique digital signature 132 is sent via "snail mail" to the remote user's personal home address. Next, the remote user logs in to Java based voting application interface residing on the WWW server 424. The WWW server 424 then presents a number of prompts (e.g., check boxes) and voting information (such as candidate or referendum information). Once the remote user has sufficiently responded to the prompts, the responses are validated to ensure that logical constraints have not been violated (e.g., selecting more than one candidate for a particular elected position). After the responses are validated, the WWW server 424 interface prompts the remote user for their unique digital signature 132. With the unique digital signature 132, the WWW server 424 also accepts the remote user's responses to the polling questions. The unique digital signature 132 is authenticated at step 708 (see FIG. 6) and, if accepted, is the remote user's selections are forwarded to an automated processing machine, such as computer 404. The computer 404 processes the remote user's voting selections at step 716 and, according to one embodiment, coordinates a

reply message back to the remote user informing the remote user that the remote user's votes were recorded. The reply message can also contain the remote user's selections/votes.

[0069] If the unique digital signature 132 was discarded at step 708, one of at least three responses (step 724) is appropriate. One response is to completely disregard the failed unique digital signature 132 and voting responses and delete them from the unique digital signature system 400. Another response is to record the failed attempt in a failure log file that a system administrator can later analyze for unusual activity. A third response is to record the failed unique digital signature 132 in the failure log file and notify the remote user of the failure (for example, including the voter's response so that the remote user can reattempt the voting process without tediously reviewing each question.)

[0070] According to one embodiment, the steps for generating an authenticating a unique digital signature 132 are performed by a computer program running on a stand-alone server (e.g., 428), or in an add-on software component in servers 424 or 408.

[0071] In one embodiment, the instructions for performing the methods and techniques described herein (the computer program) can be stored on a computer readable medium, such as an electromagnetic storage device (e.g., a floppy disk, a magnetic tape, a hard-disk drive, or other persistent memory device), or an optical data storage medium (e.g., a CD-ROM). Generally, prior to execution of the sequences of instructions, the sequences of instructions are copied from a non-volatile computer readable medium (e.g., the hard-disk drive) to a volatile source (e.g., random access memory) and are executed from the volatile computer readable medium. For purposes of

explanation the methods and techniques described herein are performed by the authenticated message server 428. Where the actual functionality is performed, that is on which piece of hardware, is not important for purposes of this description. For example, server 308 can be configured to perform the functionality of both a regular message server and a unique digital signature message server, wherein the message server automatically replaces the sender's e-mail address with the unique digital signature in response to a prefix before the desired recipient's address (e.g., "onetime.jenny@mailer.com").

[0072] In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will be evident, however, that various modifications and changes can be made thereto without departing from the broader spirit and scope of the invention. For example, larger or smaller system keys 108 (e.g., 48 or 128 bit system keys) can be employed. Further, the adapted digital signature 144 can be truncated in order to not exceed the boundaries of the address field for an electronic message. Further still the authenticated message server functionality can be incorporated into a message server (e.g. server 308 or server 408), rather than being a stand-alone device. The specification and drawings are, accordingly, to be regarded in an illustrative, rather than a restrictive sense.